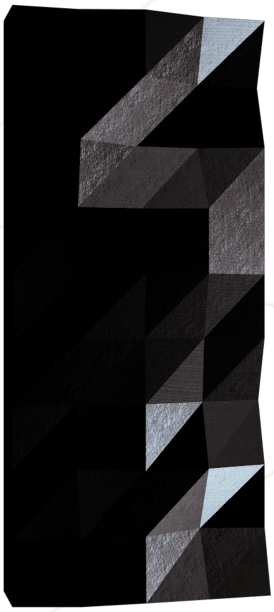


Piscine datascience - 0

Creation of a DB

Summary: Today, you will discover the creation of a DB

Version: 1.2

Contents

Chapter I

General rules

- You have to render your modules from a computer in the cluster either using a virtual machine:
    - You can choose the operating system to use for your virtual machine
    - Your virtual machine must have all the necessary software to realize your project. This software must be configured and installed.
- Or you can use the computer directly in case the tools are available.
    - Make sure you have the space on your session to install what you need for all the modules (use the goinfre if your campus has it)
    - You must have everything installed before the evaluations
- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and will receive a `0` during the evaluation.
- We encourage you to create test programs for your project even though this work **won't have to be submitted and won't be graded**. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.
- Submit your work to your assigned git repository. Only the work in the git repository will be graded. If Deepthought is assigned to grade your work, it will be done after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.
- By Odin, by Thor ! Use your brain !!!

Chapter II

Introduction

In the next two modules, you will see the role of a data engineer.

This second step is important to understand. The data engineer "cleans" the data and transforms it in order to have data ready to be analyzed by analysts/data scientists.

The next module involves data cleansing. This second step is important to understand the data engineer "cleans" the data and transforms it. The objective is to have data ready to be analyzed by analysts/data scientists.

We are at the end of February 2023, it's your first day in a company selling items on the Internet. Before leaving on a trip your boss gives you the sales of the last 4 months. You will have to exploit them and propose solutions to increase the turnover of the company.

⚠ **Be careful** with this "piscine". Even if you manage to validate a module, you may be stuck later if you haven't cleaned up or stored your data properly.

Chapter III

Exercise 00

Exercise 00
Exercice 00 : Create Postgres DB
Turn-in directory: *ex*00/
Files to turn in: `docker-compose.yml` *or* `setup.sh` *or* `VM-instructions.txt`
Allowed functions: `All`

For this exercise, you can use PostgreSQL directly if it is installed on your campus machine or on a VM. Otherwise, you must use Docker Compose.

- The username must be your student login.
- The name of the database must be `piscineds`.
- The password must be `mysecretpassword`.

We must be able to connect to your PostgreSQL database with the following command:

```
psql -U your_login -d piscineds -h localhost -W
mysecretpassword
piscineds=#
```

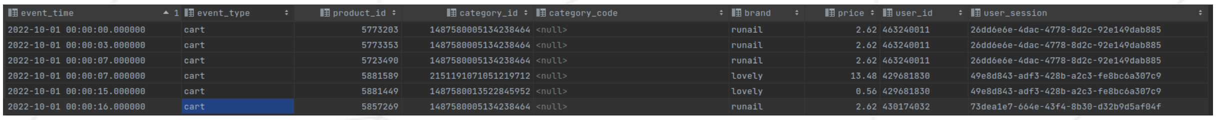
If you choose to use Docker, your setup must follow the same standards and good practices as required in the Inception project.

Chapter IV

Exercise 01

	Exercise 01
Exercice 01 : Show me your DB	
Turn-in directory: *ex*01/	
Files to turn in:	
Allowed functions: `pgAdmin, Postico, DBeaver or any other tool of your choice`	

- Find a way to easily visualize your database using a software tool.
- The chosen software should allow you to browse and manipulate data easily, especially using record IDs.



event_time	event_type	product_id	category_id	category_code	brand	price	user_id	user_session
2022-10-01 00:00:00.000000	cart	5773203	1487580005134238464	<null>	runail	2.62	463240011	26dd6e6e-4dac-4778-8d2c-92e149dab885
2022-10-01 00:00:03.000000	cart	5773353	1487580005134238464	<null>	runail	2.62	463240011	26dd6e6e-4dac-4778-8d2c-92e149dab885
2022-10-01 00:00:07.000000	cart	5723490	1487580005134238464	<null>	runail	2.62	463240011	26dd6e6e-4dac-4778-8d2c-92e149dab885
2022-10-01 00:00:07.000000	cart	5881589	2151191071051219712	<null>	lovely	13.48	429681830	49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2022-10-01 00:00:15.000000	cart	5881449	1487580013522845952	<null>	lovely	0.56	429681830	49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2022-10-01 00:00:16.000000	cart	5857269	1487580005134238464	<null>	runail	2.62	430174032	73dea1e7-664e-43f4-8b30-d32b9d5af04f

Chapter V

Exercise 02

	Exercise 02
Exercice 02 : First table	
Turn-in directory: *ex*02/	
Files to turn in: `table.*`	
Allowed functions: `All`	

- Create a PostgreSQL table using the data from a CSV file located in the `customer` folder. The table must be named after the CSV file (without the file extension), e.g., `data_2022_oct`.
- The column names must exactly match the ones in the CSV file, and their data types must be appropriate. You must use **at least six different data types**.
- A `DATETIME` column as the **first column** is mandatory.

> `Be careful: PostgreSQL data types are not exactly the same as those in MariaDB.`

Chapter VI

Exercise 03

Exercise 03
Exercice 03 : Automatic table
Turn-in directory: *ex*03/
Files to turn in: `automatic_table.*`
Allowed functions: `All`

- You are at the end of February 2022. By now, you should be able to create tables from CSV files manually.
- Now, your task is to automatically retrieve all CSV files from the `customer` folder and create a table for each one. Each table must be named after the corresponding CSV file, without its extension. For example: `data_2022_oct`.

Below is an example of the expected directory structure:

```
$> ls -alR
total XX
drwxrwxr-x 2 eagle eagle 4096 Fev 42 20:42 .
drwxrwxr-x 5 eagle eagle 4096 Fev 42 20:42 ..
drwxrwxr-x 2 eagle eagle 4096 Jan 42 20:42 customer
drwxrwxr-x 2 eagle eagle 4096 Jan 42 20:42 items


./customer:
total XX
drwxrwxr-x 2 eagle eagle 4096 Fev 42 20:42 .
drwxrwxr-x 5 eagle eagle 4096 Fev 42 20:42 ..
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 data_2022_dec.csv
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 data_2022_nov.csv
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 data_2022_oct.csv
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 data_2023_jan.csv

./items:
...
```

Chapter VII

Exercise 04

	Exercise 04
Exercice 04 : Items table	
Turn-in directory: *ex*04/	
Files to turn in: `items_table.*`	
Allowed functions: `All`	

- You must create a table named `items` using the column names provided in the `item.csv` file.
- The table must contain at least **three different data types**.

Below is an example of the expected directory structure:

```
$> ls -alR
total XX
drwxrwxr-x 2 eagle eagle 4096 Fev 42 20:42 .
drwxrwxr-x 5 eagle eagle 4096 Fev 42 20:42 ..
drwxrwxr-x 2 eagle eagle 4096 Jan 42 20:42 customer
drwxrwxr-x 2 eagle eagle 4096 Jan 42 20:42 items

./customer:
...

./items:
total XX
drwxrwxr-x 2 eagle eagle 4096 Fev 42 20:42 .
drwxrwxr-x 5 eagle eagle 4096 Fev 42 20:42 ..
-rw-rw-r-- 1 eagle eagle XXXX Mar 42 20:42 items.csv
```

Chapter VIII

Submission and Peer-Evaluation

Submit your assignment in your `Git` repository as usual. Only the work contained within your repository will be evaluated during the defense.

Make sure to double-check the names of your folders and files to ensure they match the required structure.

> The evaluation process will take place on the computer of the group being evaluated.